

CLAIMS:

1. A data processing apparatus comprising
 - at least one processing means being capable of providing data for further processing by the same or other processing means,
 - a queue structure comprising at least two branches between a producer task 5 performed by a first processing means and a number of consumer tasks executed by at least a second processing means,
 - a memory means for storing data to be accessed by said consumer tasks, said memory means being shared between said at least two branches,
 - a branch record means comprising a primary branch record for a primary 10 branch between said producer task and a first consumer task and secondary branch records for secondary branches between said producer task and further consumer tasks, said branch records storing a pointer to the same location of said memory means and a reference to the next branch so as to obtain a linked list of branch records.
- 15 2. A data processing apparatus as claimed in claim 1, wherein said secondary branch records further store a reference to the previous branch so as to obtain a double-linked list of branch records.
- 20 3. A data processing apparatus as claimed in claim 1, wherein each said branch record comprises a writer pointer denoting the producer task's position in the queue and a reader pointer denoting the consumer task's position in the queue, said writer pointer being identical for all branch records.
- 25 4. A data processing apparatus as claimed in claim 1, wherein each said branch record comprises a writer counter denoting the producer task's position in the queue and a reader counter denoting the consumer task's position in the queue, said writer counter being identical for all branch records.

5. A method of synchronizing at least two processing means in a data processing apparatus, at least one of which being capable of providing data for further processing by other processing means, said method comprising the steps of:

- defining a queue structure comprising at least two branches between a producer task performed by a first processing means and consumer tasks executed by at least a second processing means,
- sharing a memory means for storing data to be accessed by said consumer tasks between said at least two branches,
- defining a branch record means comprising a primary branch record for a primary branch between said producer task and a first consumer task and secondary branch records for secondary branches between said producer task and further consumer tasks, said branch records storing a pointer to the same location of said memory means and a reference to the next branch so as to obtain a linked list of branch records.

15 6. A method as claimed in claim 5,

further comprising the steps of:

- reading a writer pointer or writer counter from said primary branch record,
- reading reader pointers or reader counters from said secondary branch records in the linked list,
- comparing said reader pointers or reader counters with said writer pointer or writer counter to determine the amount of free buffer space in the individual branches available for writing by the producer task,
- incrementing said writer pointer or writer counter in the primary queue record after writing a new data item in the queue by the producer task,
- updating the value of the writer pointer or writer counter in the secondary branch records, and
- signaling to the consumer tasks attached to the queue the change in the queue fullness and unblocking them in case they are in a blocked state.

30 7. A method as claimed in claim 6,

wherein the actual amount of empty buffer space available in the entire queue is indicated by the minimum amount of empty buffer space over all the branches and wherein the producer task is blocked when any of the branches is full.

8. A method as claimed in claim 5,

further comprising the steps of:

- reading a writer pointer or writer counter and a reader pointer or reader counter from the associated branch record to determine the number of available data items in
5 the queue available for reading in a particular branch by the associated consumer task.

- incrementing said reader pointer or reader counter in the associated branch record after reading a data item from the queue by a particular consumer task, and
- signaling to the producer task a change in the queue fullness, thereby
unblocking the producer task in case it is in a blocked state.

10

9. A method as claimed in claim 8,

wherein the consumer task is blocked when its associated branch is empty.

15

10. A method as claimed in claim 5,

wherein a secondary branch is dynamically added to said queue structure by copying the primary branch's branch record into the new secondary branch's branch record and by adding a reference to the new secondary branch to the previously last branch's branch record.

20

11. A method as claimed in claim 5,

wherein a secondary branch is removed from said queue structure by removing its secondary branch record from said queue record means and by updating the reference to the next branch stored in its predecessor branch's branch record so as to refer to the removed branch's successor branch and by updating the reference to the previous branch stored in its successor branch's branch record so as to refer to the removed branch's predecessor branch.

25

12. A method as claimed in claim 5,

wherein the primary branch is removed from said queue structure by removing its primary branch record from said queue record means and by deleting a reference to the primary branch record stored in the first secondary branch's branch record and by informing the producer task of the deletion of the primary branch and the fact that the first secondary branch has now become the primary branch.